Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 3 (2011) 110–114

Procedia Computer Science

www.elsevier.com/locate/procedia

WCIT 2010

Extending ETL framework using service oriented architecture

Mohammed M I Awad [a] *, Mohd Syazwan Abdullah [a], Abdul Bashah Mat Ali [a]

[a] *Division of Applied Science, College of Arts and Sciences, Universiti Utara Malaysia,06010 Sintok, Kedah, Malaysia.*

Abstract

Extraction, Transformation and Loading (ETL) represent a big portion of a data warehouse project. Complexity of components extensibility is a main problem in the ETL area, because ETL components are tightly-coupled to each others in the current ETL framework. The missing extensibility feature causes impediments to add new components to the current ETL framework; to meet special business needs. This paper shows how to restructure the current ETL framework based on Service Oriented Architecture (SOA) to be easier to extend. This restructuring solution distributes the ETL into interoperable components. The distribution of Extraction, Transformation and Loading components while keeping interoperability amongst them; can be achieved by SOA. A Classified-Fragmentation component to enhance the report generation speed is added to the new framework; as a proof of the extensibility concept. The result of this work is an extensible ETL framework including Classified-Fragmentation component as an extension.

Selection and/or peer-review under responsibility of the Guest Editor.

Keywords: Data Warehouse; ETL; SOA; Extensibility; Distribution; Interoperability; Tight-Coupling; Loose-Coupling.

1. Introduction

Extraction, Transformation, and Loading processes play a central role in the data warehouse solutions. In addition, ETL is considered the core component of a successful data warehouse system [2]. It physically integrates data from multiple heterogeneous sources and stores it in a central repository referred to as data warehouse [3]. Due to its importance and high cost, many research projects are carried out to enhance ETL framework. Lots of those projects in the last few years have concentrated on Real-Time data warehousing; to solve the periodic Extraction, Transformation, and Loading problems [4]. On the other hand, there are no sufficient research works done to eliminate the difficulties in this field regarding extending ETL components; to suite additional special business needs [3,5]. For instance, many companies need to add special components to ETL; to solve special problems in their own business. Those companies face impediments because ETL components are tightly-coupled. In addition, due to the complexity, long learning curve of the available ETL tools, and difficulty to achieve extensibility in terms of additional functionalities; many organizations prefer to turn to in-house development to perform ETL tasks, which increases the project effort. Based on the explored problem of ETL framework, an enhancement to the current ETL framework based on SOA; by including the features of component distribution and interoperability addresses the extensibility problem [1, 6].

In this paper, Classified-Fragmentation component is added as an extension to the enhanced ETL framework to solve the relatively low speed report-generation of data warehouse projects. This component is important for performance issues, because

* Mohammed M I Awad. Tel.: +6-012-256-8602.
E-mail address: moh108@yahoo.com.

1877-0509
doi:10.1016/j.procs.2010.12.019

data volumes are growing at a significant pace, which makes report generation relatively slow due to the massive amount of data [7]. Some implementations of the ETL framework like Pentaho Open Source Business Intelligence [8] include similar fragmentation components. However, the fragmentation feature of those implementations is tightly-coupled in the ETL tool and it is not based on distribution and interoperability standards. Furthermore, those types of fragmentations target mainly to enable the fact and the dimension tables in the data warehouse to be separated among a cluster of servers. That belongs to the physical (hardware) solution of the performance problem, but this paper concentrates on adding a software-based and loosely-coupled Classified-Fragmentation component to the ETL framework, as an extension to it; to prove the availability of the extensibility of the framework and to meet some special fragmentation needs of an organization. The ETL framework restructuring based on SOA; is explored in sections 2 and 3 by exploring the current ETL framework and the new one resulted from this research, while the Classified-Fragmentation extension component is explored in sections 4 and 5.

2. Current ETL Framework

According to [3,5], the traditional ETL framework has common tightly-coupled functionalities. Those functionalities, concepts behind them, and relationships between them can be concluded in one framework diagram as shown in Fig. 1. In the data layer, the data stores that are involved in the overall process are depicted. On the left side, the original data providers (typically, relational databases and files) are shown. The data from these sources is extracted (as shown in the upper left part of Fig. 1) by Extraction routines. Then, this data is propagated to the Data Staging Area (DSA) where it is transformed and cleaned before being loaded to the data warehouse. The data warehouse repositories are depicted in the right part of Fig. 1 and comprise the target data stores. Eventually, the data loading to the central warehouse is performed through the loading routines depicted on the upper right part of Fig. 1.

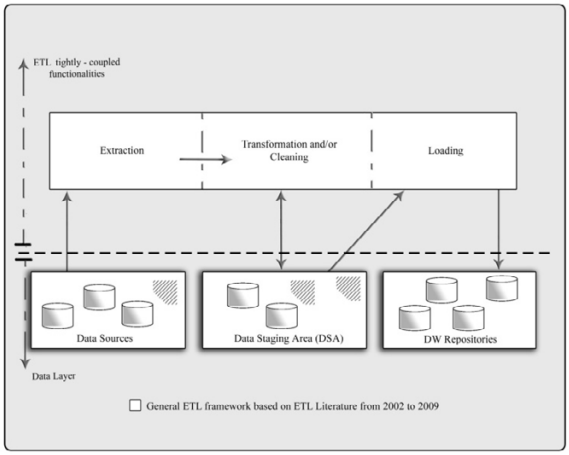

Fig. 1: Traditional ETL framework

3. ETL Framework with Interoperable Distributed Components

This section briefs the enhanced ETL framework. Fig. 2 illustrates this framework, which is based on SOA. In the data layer of Fig. 2, the data stores are exactly similar to those available in Fig 1 of the traditional framework. The business layer of Fig. 2 that is built based on SOA; includes four main parts which are:

A. *Service Orchestration Point (also called Directory Service or Service Registry):* It describes the services available in its domain which are Extraction, Transformation, and Loading. Those three services are called Service Providers and register themselves in the Orchestration Point.

B. *Service Providers:* each of them is a component that performs a service in response to a consumer request. The framework has three Service Providers which are Extraction, Transformation, and Loading services.

C. *Service Consumers:* each of them is a component that consumes the result of a service supplied by a provider. The main Service Consumer in the framework is the client that represents ETL administrators. In addition, the three Service Providers can be Service Consumers to other services. For example, the Transformation service can request some functions to be done by the Extraction service in case that the Extraction and the Transformation are executed in one patch.

D. *Service Interface:* it defines the programmatic access of the three services, and establishes the identity of the service and the rules of the service invocation.

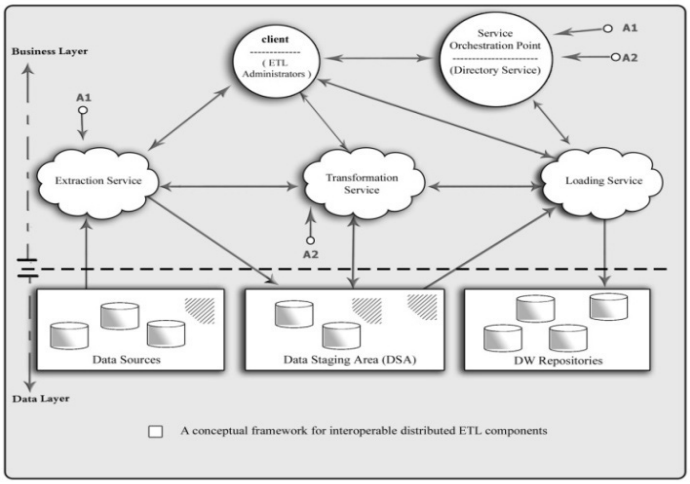

Fig. 2: A framework for interoperable distributed ETL components

A simple flow diagram is shown in Fig. 3 and described below; to clearly show the flow of actions when a client demands an execution of an ETL functionality. When a client demands to consume (execute) a certain ETL service, the Orchestration Point starts with a “receive” activity in which it receives the client request. Then, proceeds with invoking the suitable ETL service(s) and finishes by replying back to the client. An Orchestration Point Process typically interacts with one or more ETL web services. These ETL web services are called partner services or external service.

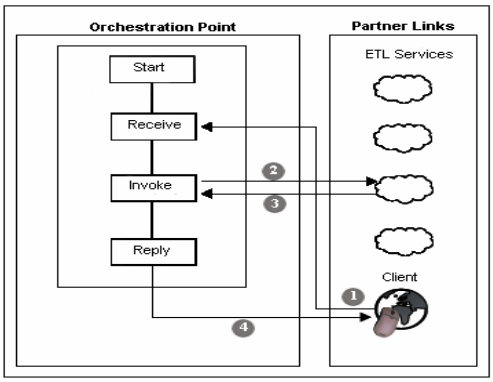

Fig. 3: Flow diagram for steps to consume an ETL service by a client [1]

4. Adding a Classified-Fragmentation Component as an Extension to the restructured Framework

A Classification-Fragmentation component is added as an additional component to the enhanced ETL framework; to speed up the report generation, and to show the simplicity and flexibility in adding any new component as an extension to the restructured ETL framework; without affecting other components. As shown in Fig. 4, the framework includes the Classification-Fragmentation component as an extension. After doing the prototype of the Classified-Fragmentation component (explored in section 5) based on the framework of Fig. 4, it is proved that by following the SOA concept, any other component can be added to the framework without any complications. Section 5 explains the Classified-Fragmentation component.

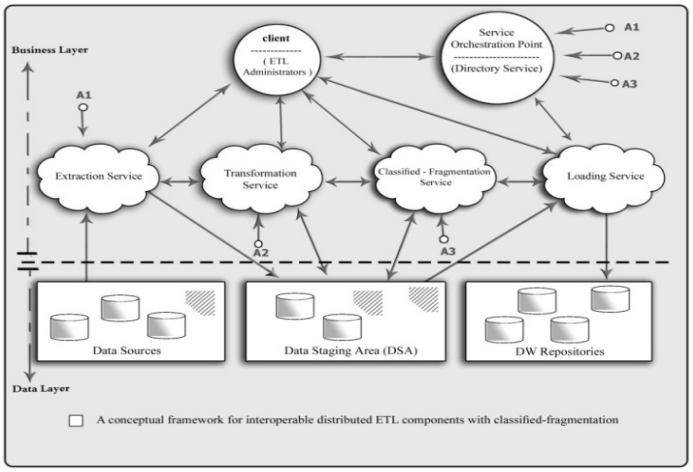

Fig. 4: A framework for interoperable distributed ETL components with classified - fragmentation

5. Classified-Fragmentation Component

The Classified-Fragmentation is added to enhance the speed of report generation of a data warehouse project. It classifies and fragments data into groups based on the type of data. After executing the classification and fragmentation processes based on data types, several tables are created. Then, the classified data is stored into these tables. Afterwards, the statistical reports are generated based on the fragmented fact and dimension tables of the data warehouse according to the required data type. That

decreases the processed data during the report generation. For testing purposes, A report generation speed test is done using Apache JMeter tool [9]. Testing is done to calculate the time consumed to generate a report in data warehousing. This testing is done 9 times using fragmented data and 9 times using un-fragmented data, while different numbers of concurrent users are considered. A random report is generated using an amount of records from 100,000 to 300,000, and from one to three concurrent users. The time consumed for each of fragmented and un-fragmented data is shown in Table 1 in milliseconds (ms). As shown in Table 1, it is clear that the time required to generate a report in case of fragmented data is less than the time needed for un-fragmented data.

Table 1: Time deference between Fragmented and Un-Fragmented data for report generation.

	No. of Records	No. of Users	Fragmented data (ms)	Un-fragmented data (ms)
1	100,000	1	188	297
2	100,000	2	192	207
3	100,000	3	255	364
4	200,000	1	359	515
5	200,000	2	369	520
6	200,000	3	369	643
7	300,000	1	516	688
8	300,000	2	531	719
9	300,000	3	625	815

6. Discussions and Conclusions

Overall, the loosely-coupled services (components) in the ETL framework are typically more flexible than those of tightly-coupled ETL framework. In the traditional framework, the ETL components are tightly-coupled to each other, sharing semantics, libraries, and often sharing state. This makes it difficult for the application to evolve and to adapt with the ever changing business requirements. The asynchronous nature of loosely-coupled framework services discussed in this paper allows applications to be more flexible, and easy to incorporate additional new requirements. This is done by restructuring the ETL framework to include an extra component without adding additional complexities, which is the Classified-Fragmentation service. Exploiting this, the restructured ETL framework was developed based on the SOA, which can also be extended in the future by adding extra components to suit enterprises' new business needs.

References

1. Salter, D. and F. Jennings, *Building SOA-Based Composite Applications Using NetBeans IDE 6.* 2008.
2. Zhou, X., et al., *Building Clinical Data Warehouse for Traditional Chinese Medicine Knowledge Discovery.* International Conference on BioMedical Engineering and Informatics,IEEE, 2008
3. Jörg, T. and S. Deßloch, *Towards Generating ETL Processes for Incremental Loading.* ACM, 2008.
4. Santos, R.J. and J. Bernardino, *Real-Time Data Warehouse Loading Methodology.* ACM, 2008.
5. Tziovara, V., P. Vassiliadis, and A. Simitsis. *Deciding the physical implementation of ETL workflows*. 2007: ACM.
6. Wang, C. and S. Liu, *SOA Based Electric Power Real-time Data Warehouse.* Workshop on Power Electronics and Intelligent Transportation System,IEEE, 2008.
7. Mundy, J., W. Thornthwaite, and R. Kimball, *The Microsoft data warehouse toolkit: with SQL Server 2005 and the Microsoft Business Intelligence toolset*. 2006: Wiley Pub.
8. Pentaho. *Pentaho Business Intellegence*. 2009 [cited 2/7/2009]; Available from: http://www.pentaho.com/?_kk=pentaho&_kt=8af845c0-288a-4d60-af2a-a8c4a85a102f&gclid=CPL94emNw5wCFRwpawodvkiWnA.
9. Apache. *Apache JMeter*. 2010 [cited 22/9/2009]; Available from: http://jakarta.apache.org/jmeter/.